A Study on The Conversion Factor between Heterogeneous DBMS for Cloud Migration

Joonyoung Ahn[1], Kijung Ryu[1*], Changik Oh[1], Taekryong Han[1], Heewon Kim[2] and Dongho Kim[2]
[1] Department of IT Policy and Management, Graduated School of Soongsil University
Seoul, 06978 Republic of Korea
[e-mail: corea003@gmail.com, yoo123111@gmail.com, lowerright@gmail.com, taekryong.han@gmail.com]
[2] Global School of Media, Soongsil University, Seoul, 06978 Republic of Korea
[e-mail: hwkim@ssu.ac.kr, dkim@ssu.ac.kr]
[*]Corresponding author: Kijung Ryu

Received April 8, 2024; accepted June 3, 2024;
published August 31, 2024

Abstract

Many legacy information systems are currently being clouded. This is due to the advantage of being able to respond flexibly to the changes in user needs and system environment while reducing the initial investment cost of IT infrastructure such as servers and storage. The infrastructure of the information system migrated to the cloud is being integrated through the API connections, while being subdivided by using MSA (Micro Service Architecture) internally. DBMS (Database Management System) is also becoming larger after cloud migration. Scale calculation in most layers of the application architecture can be measured and calculated from auto-scaling perspective, but the method of hardware scale calculation for DBMS has not been established as standardized methodology. If there is an error in hardware scale calculation of DBMS, problems such as poor performance of the information system or excessive auto-scaling may occur. In addition, evaluating hardware size is more crucial because it also affects the financial cost of the migration.

CPU is the factor that has the greatest influence on hardware scale calculation of DBMS. Therefore, this paper aims to calculate the conversion factor for CPU scale calculation that will facilitate the cloud migration between heterogeneous DBMS. In order to do that, we utilize the concept and definition of hardware capacity planning and scale calculation in the on-premise information system. The methods to calculate the conversion factor using TPC-H tests are proposed and verified. In the future, further research and testing should be conducted on the size of the segmented CPU and more heterogeneous DBMS to demonstrate the effectiveness of the proposed test model.

Keywords: Cloud Migration, Scale Calculation, DBMS, Conversion Factor, TPC-H

A preliminary version of this paper was presented at ICONI 2023, and was selected as an outstanding paper.

http://doi.org/10.3837/tiis.2024.08.022 **ISSN : 1976-7277**

1. Introduction

According to many studies and recent reports, the adoption of cloud computing has increased during COVID-19 pandemic[1]. This is because the importance of cloud technology became more apparent in the pandemic environment. But despite the high level of interest in cloud deployment, there are still a lot of the burden and concern[2]. After the pandemic, the number of cloud transition has been reduced a little bit as many digital transformation projects slowed down, but many companies are still moving forward with the expectation for cost reduction[3][4]. Existing on-premises systems tend to prioritize stability and thus secure even unnecessary resources in advance, which often wastes money[5]. If it is determined that the cost of cloud transition is lower than the cost of maintaining existing IT systems, cloud transition will be more accelerated[6]. One of the biggest issues when converting to the cloud is to switch to the cloud while using the capacity and settings designed for the on-premise system because of fast transition and difficulty in design, even though redesigning is necessary for cost efficiency. In other words, in order to convert an existing IT system to the cloud, a design that considers the cloud environment must be preceded, but it focuses only on the transition to the cloud and does not properly calculate the size, which is a key indicator for successful migration[5][6].

Cloud migration in terms of DBMS is also proceeding without accurate calculation. Scale calculation is important when converting to the cloud, especially if different DBMS engines are installed on-premises and the cloud, due to the different costs.

Fig 1. is a cloud migration process that consists of Enable – Evaluate – Design, Develop & Standardize – Deploy – Optimize[7]. The scale calculation of heterogeneous DBMS in cloud migration, which is focused on in this paper, should be defined at the Enable stage[8]. If the analysis and definition of the existing on-premises system are not performed in advance, problems may arise in the Analysis and Proof of Concept stage and cannot proceed to the subsequent stages. Migration without scale calculation is not a true migration, and it results in only moving the location to the cloud. Thus, it is hard to benefit from the advantages of operational efficiency, cost efficiency, and shortening the development period due to the cloud migration[7][8].

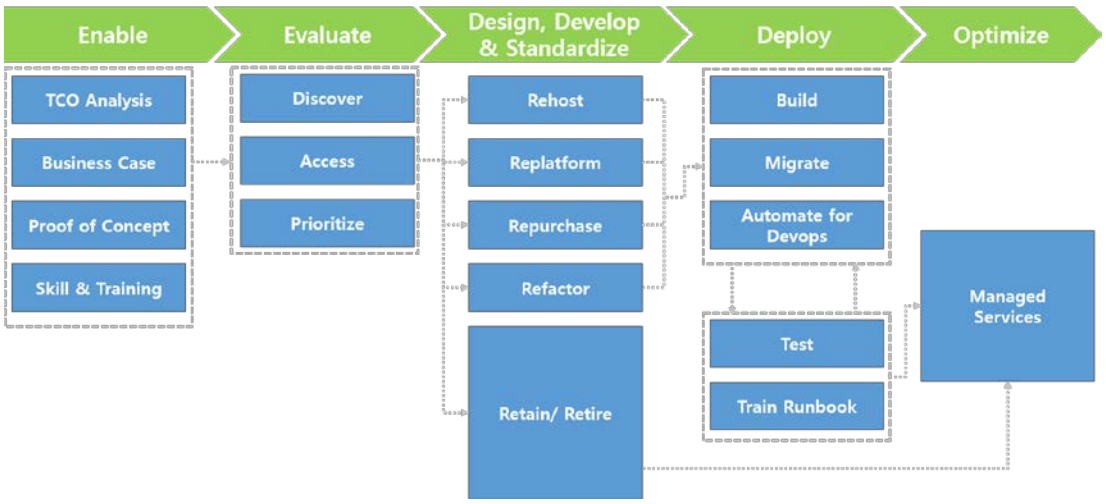

Fig. 1. Cloud migration process

If a problem occurs during the DBMS migration to the cloud, it may be solved while auto-scaling operates in the Optimize step. However, in this case, excessive load cannot be avoided at the start of the Managed Service, and outages may occur in severe cases. This kind of scale calculation error should be regarded as a failure to migrate to the cloud from the perspective of operational efficiency and cost efficiency[8].

The purpose of this paper is to propose a new method for determining the appropriate CPU core for optimal DBMS performance by performing TPC-H tests on various heterogeneous DBMSs. The characteristics of each query constituting the TPC-H test can also be reflected to determine the appropriate DBMS for each task.

The composition of this paper is as follows. Chapter 2 describes the concept and calculation method of scale calculation in the on-premises information system as related research. Chapter 3 proposes the conversion factor calculation technique and verification method for the scale calculation of heterogeneous DBMS. Chapter 4 confirms the effectiveness of the proposed technique through experiments. Finally, Chapter 5 concludes with a description of conclusions and future studies.

2. Related Research

2.1 Definition of Scale Calculation

The terms of capacity management, capacity planning, and scale calculation are used interchangeably without clear distinction. However, there are clear differences between these terms. First, in order to define the exact concept of hardware scale calculation, the definitions and concepts of capacity management, capacity planning, and scale calculation are given as follows[9][10].

Capacity management can be defined as establishing a capacity plan necessary to meet business requirements and to balance between cost and capacity. Capacity management focuses on continuous management and includes not only information system resources within the organization, such as systems and networks, but also enterprise resources[9][10].

Capacity planning can be understood as a plan to determine the performance required by the system based on the schematic system architecture and application tasks. This capacity planning is determined through the peak rate and margin rate of the information system, which are calculated based on the shape, operation characteristics, and the number of users of the client applications[9][10].

On the other hand, scale calculation refers to the conversion of basic capacity and performance requirements into system requirements. The factors determined at the time of scale calculation include the CPU type or the number of servers, disk size or shape, memory size, and network capacity[9].

In summary, scale calculation uses a mathematical methodology to calculate the size of the information system, and given the architecture of the information system, it can be viewed as a sublevel concept of capacity planning and capacity management that determines performance and management requirements[9][10].

In scale calculation, performance evaluation is a very important process in the design and implementation stage of software systems such as DBMS or middleware, and the benchmark methods developed by public organizations are widely used[10].

Specifically, in the case of DBMS, the benchmark of TPC (Transaction Processing Performance Council) is used, and the benchmark test method of SPEC (Standard Performance Evaluation Corporation) is applied to middleware[10]. Using this standardized benchmark

method, it is possible to evaluate the performance of the system on an objective and consistent basis[9].

When performing system performance evaluation and scale calculation, it is important to comprehensively consider the characteristics of the system architecture and workload, not just the function of the software. This reflects various situations that the system will experience in the actual operating environment and enables accurate performance prediction. In particular, the scale calculation of the DBMS CPU should vary depending on the characteristics of the workload that the server must process. The type, size, and complexity of transactions handled by DBMS greatly affect the server's CPU usage[10][11].

For example, a high performance transaction processing system that needs to process a large amount of data in real time will require high CPU performance, and a system that focuses on data retrieval will require relatively low CPU performance. This approach that considers the characteristics of the system architecture and workload in the scale calculation process meets the various requirements that the system may face in the future while ensuring efficient resource usage.

2.2 Three Methods of Scale Calculation

The scale calculation of an information system is a matter of future prediction of usage, and the accuracy depends on the calculation method[10]. As summarized in **Table 1**, three types of methods are used to calculate the size of the information system: the Calculating Method, the Referencing Method, and the Simulation Technique[9].

Table 1. Methods of scale calaulation

Classification	Concept	Advantage	Disadvantage
Calculating Method	- Identifying factors for scale calculation - Calculating capacity - Applying calibration values	- Clear basis for scale calculation - Simple calculation method	- Errors according to calibration values
Referencing Method	- Calculation referring to similar systems	- Low possibility of error in scale calculation	- Difficulty in presenting evidence
Simulation Technique	- Simulation after workload modeling	- Accurate scale calculation	- Time and cost consuming

The first, calculating method has the advantage of calculating the capacity value based on the number of users and applying the calibration value to provide an accurate basis. However, if the calibration value is incorrect, there is a disadvantage that the results may differ a lot from the required capacity[9].

The second, referencing method is to calculate by referring to the size of similar systems based on basic data such as workload. When the referenced information system is stable, it can show high reliability, but it is difficult to present the calculation basis because it is just a referencing method rather than a calculation method[9].

The third, simulation technique is a method of modeling and simulating the workload for the target task and calculating the scale, which can obtain accurate results compared to the referencing method, but the calculation process is expensive and time consuming[9].

In scale calculating of the current information system, most of them use the referencing method, and calculating methods are also used in some cases. But there are many problems when building the system due to the inability to accurately calculate the size. In the cloud

migration, a dedicate scale calculation guide is not presented, so the existing scale calculation guide for on-premises construction is currently followed[11]. Using the referencing method in the migration process is the same as using the system that will be used in the cloud as a tester. In other words, it is to test the operation after applying it to B system without changing the settings and data of the existing A system. If cost and time are not taken into account, testing the new system using the existing system is a good way to check the performance in the real environment. But it should be used carefully because it can lead to dangerous situations that affect existing system.

2.3 Test Model for Migration

2.3.1 Selection of Benchmark Test Models

The TPC is a committee established in the United States in 1998 and is an organization that sets standards for the processing performance of transactions. The TPC presented 18 benchmark tests, and currently 10 benchmark tests are actually used[12].

In the past, TPC-C, a benchmark test for OLTP (Online Transactional Processing), was used as a scale calculation method in the operating environment of the on-premises information system[13], and benchmarks for DSS (Decision Support System) have also been widely used recently. It is not clearly determined which of the two benchmark tests is better than the other. But DSS accesses more databases than OLTP, and because the transaction itself consists of large and long complex queries, it typically takes longer to execute and provides a harsh environment where DBMS_LOCK occurs a lot[14].

TPC-D is a DSS benchmark model, which can reflect recent information system tendencies with complex queries[14]. In this paper, we use TPC-H, which is a DSS benchmark model improved from TPC-D with added power and throughput tests.

TPC-H's 22 power test queries have their own business characteristics, and they can be divided into three main types: 12 update tasks, 5 insert tasks, and 5 writing tasks. [12].

2.3.2 Measurement of Performance

The TPC-H query power test results are used to calculate the processing power according to the database size. The processing time of a query is to calculate a special kind of average, that is, a geometric mean, by utilizing the values of all measured intervals. In this method, after multiplying the values of all intervals, an average value is derived by taking a root corresponding to the number of multiplication results (the number of measured intervals). This approach is suitable as a mathematical method that can evenly reflect volatility when each time interval is different and the fluctuation is large. The reason for using geometric mean is that there are cases where the arithmetic mean method, which simply adds all values and divides them by the number, does not properly reflect the large difference between the intervals[12][15].

For example, when some queries are processed very quickly and some are processed relatively slowly, geometric averages fairly reflect the effects of these different processing speeds, providing more balanced averages.

This calculation method is particularly useful in benchmark measurement methods such as TPC-H. The TPC-H benchmark is used to evaluate the performance of complex database queries, and by obtaining the average value of various query processing times, the overall performance of the database system can be understood comprehensively. This is expressed as an equation as follows[12]:

$$\text{TPC} - \text{H Power@size} = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=22} RI(j,0)}} \quad (1)$$

QI(i,0) represents the query operation time within a single query of the power test. RI(j,0) represents the operation time of the query after realizing the data changed by driving the information system within the single query of the power test. Size is the database size selected for measurement and SF is defined as the corresponding scale factor[12][15].

The throughput test is used to measure the scale of the DBMS. It measures the ratio to the number of queries operated within the measurement time. 3600 is the length of an hour expressed in seconds and 22 is the number of queries in TPC-H. Since the performance results of the TPC-H benchmark are mainly expressed as the number of queries that can be processed per hour, they are calculated using this number. Ts is the total execution time taken to execute all 22 queries for a specific scale factor (SF). It is a measure of the time required for the system to process a given workload[12][15].

SF is a scale factor and represents the size of the database used in the TPC-H benchmark. The larger the SF value, the larger the size of the database, and the complexity of the test and the amount of data to be processed will increase. The scale factor is used to control the data volume in the test environment[12][15].

The throughput test is used to calculate the number (throughput) of queries that the database system can process in an hour at a given scale factor, which is expressed in the following formula[12]:

$$\text{TPC} - \text{H Through@Size} = \frac{S * 22 * 3600}{T_s} * SF \quad (2)$$

TPC-H Throughput@Size refers to the Queries per hour * Scale-Factor, and according to the TPC test guide, measurements are checked to one decimal place and then rounded up[12].

QphH@size (TPC-H composite query-per-hour performance metric) is defined as a combination of TPC-H power test results and TPC-H Throughput@Size, which can be formulated as follows[12]:

$$\text{QphH@Size} = \sqrt{Power@Size * Throughput@Size} \quad (3)$$

QphH@Size represents the number of queries per hour, where measurements are checked to a decimal place and then rounded up[12].

This paper calculates and verifies the conversion factor based on the execution time of QphH@size.

3. A Method for Calculating Conversion Factor for Scale Calculation of Heterogeneous DBMS

This paper proposes a conversion factor calculation technique for the scale calculation of heterogeneous DBMS migrated to the cloud. Migration is defined as converting an on-premises information system currently in operation to a cloud system. In this paper, the conversion factor for the scale calculation of heterogeneous DBMS is defined as "a value for

correcting the performance difference between the reference DBMS and the DBMS to be changed," As the standard product, Product O is used and it is the product operated the most currently. For example, when 8 core of CPU is allocated from product O of the current operating system, and if product P operates as 16 core after migration, the conversion factor becomes 2 in this case.

In this paper, a TPC-H test is conducted to calculate the conversion factor of DBMS before and after cloud migration. The result of the test is calculated as the query operation time. The conversion factor is calculated by dividing it into an overall conversion factor and a conversion coefficient according to three tasks classified as update, insert, and writing, so that it can be used in various situations.

3.1 Process for Calculating Conversion Factor

The method of calculating the conversion factor for conversion uses a combination of the Referencing method and Simulation technique mentioned in **Table 1** of 2.2. For the selection of the standard product, the Referencing method, which is calculated by referring to the capacity of the business system to be converted, is used. After that, comparative evaluation through simulation technique is conducted to scale calculation of the DBMS to be converted. For the performance calculation of heterogeneous DBMS, it is calculated as QphH@size of 2.4.2, and the conversion factor is derived by comparing the QphH@Size between heterogeneous products.

The performance expressed in 2.3 is a method of determining the product of DBMS and calculating the capacity, so it is based on CPU characteristics and memory characteristics. Since the CPU is the factor that has the greatest influence on the hardware scale calibration of DBMS, this paper proposes the CPU scale calibration technique.

In the scale calculation of a general information system, a product is determined and the scale calculation of CPU and memory is selected. However, this test is conducted while fixing CPU and memory between scale calculations and changing DBMS products. Based on this, QphH@size of 2.4 is derived, the conversion factor is calculated through 3.2 (4), and the average is calculated through (5).

When applying the conversion factor in this paper to an information system that migrates to the cloud, it is based on the CPU usage scale of DBMS operated by the on-premises information system, and the conversion factor is applied to scale calculation of the conversion business system.

Finally, the calculated conversion factor is verified. The work can be verified by checking the required capacity of (6) applying the conversion factor to the QphH@size for the standard.

3.2 Calculation of Conversion Factor according to Work Tasks and Overall

This experiment is derived as a proportional value of heterogeneous DBMS compared to the reference DBMS. The derived value is calculated as two types of overall conversion factor to be applied to the entire business system and tasks conversion factor.

First, in the overall scale calculation of the information system to be converted, the test operation time of 22 queries of the power test is derived as a result. Second, in the scale calculation according to the work characteristics of each unit according to the MSA (Micro Service Architecture), three conversion factors for each task described in 2.5 are derived as a result.

The calculation of the conversion factor is formulated as follows:

$$\text{Conversion Factor} = \frac{X'QphH@Size_{(x\ core)}}{O'QphH@Size_{(x\ core)}} \tag{4}$$

O'QphH@size is the scale calculation of the DBMS of the reference O product, and the result is expressed as operation time. X'QphH@size is the scale calculation of heterogeneous DBMS, and the result is expressed as operation time.

X represents the number if cores: 8 cores, 12 cores, and 16 cores. Between the calculation of the conversion factor, the same variable should be put into the reference product and the heterogeneous product subject to migration.

O'QphH@size enters the denominator as a reference, and the value of the heterogeneous DBMS subject to migration is X'QphH@size as a numerator.

$$\text{Conversion Factor} = \left\{ \begin{array}{c} \left(\frac{X'QphH@Size_{(8\ core)}}{O'QphH@Size_{(8\ core)}}\right) + \left(\frac{X'QphH@Size_{(12\ core)}}{O'QphH@Size_{(12\ core)}}\right) \\ + \left(\frac{X'QphH@Size_{(16\ core)}}{O'QphH@Size_{(16\ core)}}\right) \end{array} \right\} /3 \tag{5}$$

The average conversion factor is the average value of the conversion factor for each core, which is tested three times with 8 cores, 12 cores, and 16 cores each. The separate calculation of the conversion factor and the average conversion factor is to cope with an error in which the DBMS performance does not increase in proportion to the CPU size, but in contrast, there may be a slowdown or decrease in change when the scale calculation of a large information system is required.

The conversion factor and the average conversion factor should be calculated by applying the same procedure to the overall part and the tasks of update, insert, and writing.

Product O has a conversion factor of 1, and the closer the conversion factor of the product to be changed is to 1, the more the system can be migrated without considering additional CPU input.

3.3 Verification of Conversion Factor Results

The verification of the calculation result of the conversion factor proceeds to check whether similar results are produced in the actual test for the calculated conversion factor.

The verification method is formulated as follows:

$$X'QphH@Size_{(x\ core)} \approx O'QphH@Size_{(x'\ core)} \tag{6}$$

x' core means a core value calculated by dividing the conversion factor with respect to the core of the reference product. This is to determine the core of the reference product proportional to the core of the comparative product.

Verification confirms that similar values are obtained by comparing QphH@size(x’ core) of the O product with QphH@size(x core) of the product to be compared in (6).

This verification method should be applied equally to the overall and the three tasks: update, insert, and writing.

4. Experimental Result

4.1 Test Environment

The experiment was conducted on two commercial and three open source products.

Table 2. Test equipment information

Equipment	Detail	Specific Details	Note
CPU	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz	16 Cores 32 Threads	
MotherBoard	ASUS WS C621E SAGE		
RAM	128 GB	64GB * 2EA (2400 MT/s)	
STORAGE	1.40 TB	NVMe SSD	

The test was preformed under the same environment on one device, and the specifications of the device are summarized in **Table 2**.

The environment of the test equipment was set to 1 Socket and 100GB, and tested with each 3 types such as 8 core, 12 core, and 16 core.

4.2 Test Result

The test results are collected by 22 queries, and the sum is expressed as QphH@size.

As a result of the test, only 3 out of 5 products generated results. Product A and product M ran as a single core, and QphH@size was checked in 3 queries, but it generated timeout results in 19 queries. Looking at the results of the three products, product O and product T show similar performance, and product P shows excessive performance degradation in a specific query. The results show a significant difference between 8 core and 12 core at 16 core.

4.2.1 Overall Conversion Factor

When the result of QphH@size 16 core is expressed as O:T:P, it is 466:622:6754. If this is changed to the conversion factor of the overall, O:T:P becomes 1:1.3:14.5. And the same result was derived when the average conversion factor was calculated. In other words, when the information system operated by 16 core of the product O is converted to product T, which is a heterogeneous DBMS, it can be scaled to 20.8 core.

Table 3. Overall test result

Overall	Product O			Product T			Product P		
	16core	12core	8core	16core	12core	8core	16core	12core	8core
QphH@size	465.8	628.9	931.7	622.0	777.5	1225.4	6754.8	9456.8	13239.5
Conversion Factor	1.0	1.0	1.0	1.3	1.2	1.3	14.5	15.0	14.2
Average Conversion Factor	1.0000			1.3			14.6		

4.2.2 Conversion Factor According to Work Tasks

In order to calculate the conversion factor for work characteristics, the work characteristics of 22 queries of TPC-H were divided into three categories: update, insert, and writing.

First, 12 queries, including numbers 1 and 2, showed update task. Among the measured values for each query of QphH@size, the value of the query showing the update task is shown in

Table 4.

When the QphH@size 16 core result of update tasks is expressed as O:T:P, it is 226:458:4900. If this is changed to the update task conversion factor, O:T:P becomes 1:2:21. And the same result was derived when the average conversion factor was calculated. As a result, product P is not suitable for update tasks. In particular, product P is not suitable for query 9 that calculates the Profit Measurement by Product Type and query 18 that calculates Top Customers.

Table 4. Update tasks result

Update	Product O			Product T			Product P		
	16core	12core	8core	16core	12core	8core	16core	12core	8core
QphH@size	225.7	304.8	451.5	457.9	572.3	902.0	4900.0	6860.0	9604.0
Conversion factor	1.0	1.0	1.0	2.0	1.9	2.0	21.7	22.5	21.3
Average Conversion Factor	1.0			2.0			21.8		

Second, five queries, including numbers 3 and 4, showed insert task. Among the measured values for each query of QphH@size, the sum of the query values showing insert task is shown in **Table 5**.

When the QphH@size 16 core result of insert tasks is expressed in O:T:P, it is 153:50:1238. If this is changed to the update task conversion factor, O:T:P becomes 1:0.3:8. As a result, product P is not suitable for insert tasks. And the same result was derived when the average conversion factor was calculated. The peculiarity of insert task is that product T outperformed the standard product O, and it is the only experimental result that precedes the standard product.

Table 5. Insert tasks result

Insert	Product O			Product T			Product P		
	16core	12core	8core	16core	12core	8core	16core	12core	8core
QphH@size	153.4	207.1	306.9	53.1	66.3	104.5	1237.6	1732.7	2425.8
Conversion factor	1.0	1.0	1.0	0.3	0.3	0.3	8.1	8.4	7.9
Average Conversion Factor	1.0			0.3			8.1		

Finally, writing tasks were shown in five queries, including 5 and 11. Among the measured values for each query of QphH@size, the values of the query showing insert characteristics are summed as shown in **Table 6**.

When the QphH@size 16 core result of the writing task is expressed in O:T:P, it is 87:111:617. If this is changed to the writing tasks conversion factor, O:T:P becomes 1:1.2:7. And a similar result was derived when the average conversion factor was calculated. In the writing tasks, product P showed the lowest performance, but in query 20, which is a Promotion Potential business query, product P showed the best performance. This is a valid result to confirm that product P can also exhibit excellent performance for a single writing task.

Table 6. Writing tasks result

Writing	Product O			Product T			Product P		
	16core	12core	8core	16core	12core	8core	16core	12core	8core
QphH@size	86.7	117.0	173.3	111.1	139.0	219.0	617.2	864.1	1209.7
Conversion factor	1.0	1.0	1.0	1.3	1.2	1.3	7.1	7.4	7.0
AVE Conversion Factor	1.0			1.2			7.2		

When the QphH@size 16 core result of the writing task is expressed in O:T:P, it is 87:111:617. If this is changed to the writing tasks conversion factor, O:T:P becomes 1:1.2:7. And a similar result was derived when the average conversion factor was calculated. In the writing tasks, product P showed the lowest performance, but in query 20, which is a Promotion Potential business query, product P showed the best performance. This is a valid result to confirm that product P can also exhibit excellent performance for a single writing task.

4.3 Verification of Results

The verification of the results for the conversion factor can be calculated from the test results for product T, but in the case of product P, the conversion factor was significantly out of the range of the results in all parts of the test, so the verification work could not be carried out.

In the overall part, the conversion factor of the result of 16 core based on product O and the result of product T is 1:1.3, and the x' core is 12.3 core, a 30% reduction from 16 core of product O. The verification is conducted in the form of comparing the 12 core results of product O with 16 core of product T. In the 12 core of product O in **Table 3**, QphH@size is 628.94, and QphH@size is 622.04, indicating that the conversion factor is an applicable value. This can be checked in the same way for each task.

First, in the case of the update task, it can be seen in **Table 4**, the conversion factor of O and T products is 1:2, and the x' core is 8core, a 50% reduction from the 16core of O product. When comparing the 8core of O product with the 16core of T product, it can be seen that it also fits 451.58:457.86.

The results of insert tasks can be found in **Table 5**, and the conversion factor of product O and T is 1:0.3, and x' core becomes 16 core, a 50% increase from 8core of product O. When comparing 16core of product O with 8 core of product T, it is out of the verifiable range of 153.44:104.5. However, it can be inferred that this is correct when calculated as a ratio.

Finally, the results of writing tasks can be found in **Table 6**, and the conversion factor of product O and Product T is 1:1.2, and when comparing 12 core of product O and 16 core of product T, it is also consistent with 116.98:111.13.

5. Conclusion

With the development of cloud technology, cloud migration will proceed faster in the future, and the hardware scale calculation of DBMS acts as an important factor in migration. Moreover, cloud migration between heterogeneous DBMS will occur continuously. The CPU is the factor that has the greatest influence on the hardware scale calculation of DBMS.

This paper proposed a method of calculating a detailed conversion factor for CPU scale calculation that would facilitate the cloud migration of heterogeneous DBMS, and verified the

proposal.

Next, the conversion factor was calculated through the TPC-H test. The calculated conversion factor was calculated by dividing it into an overall conversion factor and a conversion factor according to three tasks classified as update, insert, and writing so that it can be used in various situations. Finally, the proposed calculation technique was verified by checking the conversion factor calculated by the test model.

In the future, further study on additional conversion factors is expected to contribute to the efficiency from the cost perspective and stability from an operational perspective of cloud migration.

Acknowledgement

This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2024-RS-2022-00156360)

References

[1] Kopáčková, Hana, and Sann Thawdar Htoo, "Cloud Computing Services – Emerging Trends During the Times of Pandemic," in *Proc. of 2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, pp.1-6, 2023. Article (CrossRef Link)

[2] Chen, Mei et al., "Net and configurational effects of determinants on cloud computing adoption by SMEs under cloud promotion policy using PLS-SEM and fsQCA," *Journal of Innovation & Knowledge*, vol.8, no.3, 2023. Article (CrossRef Link)

[3] Ranganathan, Chitra Sabapathy, and Rajeshkumar Sampathrajan, "Cloud Migration Meets Targeted Deadlines," in *Proc. of 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp.672-676, 2023. Article (CrossRef Link)

[4] Ahmadi, Sina, "Elastic Data Warehousing: Adapting To Fluctuating Workloads With Cloud-Native Technologies," *Journal of Knowledge Learning and Science Technology*, vol.2, no.3, pp.282-301, 2023. Article (CrossRef Link)

[5] L'Esteve, Ron C., "Influencing Change and Driving Cloud Adoption," *The Cloud Leader's Handbook: Strategically Innovate, Transform, and Scale Organizations. Berkeley*, pp.13-30, 2023. Article (CrossRef Link)

[6] Muhammad, Tayyab, "A Comprehensive Study on Software-Defined Load Balancers: Architectural Flexibility & Application Service Delivery in On- Premises Ecosystems," *International Journal of Computer Science and Technology (IJCST)*, vol.6, no.1, pp. 1-24, 2022. Article (CrossRef Link)

[7] Fahmideh, Mahdi et al., "A generic cloud migration process model," *European Journal of Information Systems*, vol.28, no.3, pp.233-255, 2019. Article (CrossRef Link)

[8] Parvathy, L. Rama, "A Comprehensive Analysis of Cloud Migration Strategies: Efficiency Comparison of Trickle, Big Bang, Refactoring, Lift and Shift, Replatforming Approaches," in *Proc. of 2023 9th International Conference on Smart Structures and Systems (ICSSS)*, pp.1-7, 2023. Article (CrossRef Link)

[9] Jonghei Ra, "A Guideline for Hardware Sizing of Information Systems," *TTA Journal*, vol.172, pp.80-82, 2017. Article (CrossRef Link)

[10] Jonghei Ra, "A Guideline for Hardware Sizing of Information Systems," *Telecommunications Technology Association*, pp.1-50, 2018. Article(CrossRefLink)

[11] Jonghei Ra, Kwangdon Choi, "A Study on the Capacity Calculation Method for Estimating the Size of Information System Introduction," *Korea Association of Information Systems*, pp.307-313, 2005. Article (CrossRef Link)

[12] Transaction Processing Performance Council, TPC Benchmark™ H Standard Specification Revision 3.0.1, 2022. Article (CrossRef Link)
[13] Min, Jae-H., Sungwoo Chang, and Kyung-shik Shin, "A Hybrid Approach to Information System Sizing and Selection using Simulation and Genetic Algorithm," *Korean Management Science Review*, vol.24, no.2, pp.143-155, 2007. Article (CrossRef Link)
[14] Elnaffar, Said et al., "Is it DSS or OLTP: automatically identifying DBMS workloads," *Journal of Intelligent Information Systems*, vol.30, pp.249-271, 2008. Article (CrossRef Link)
[15] Gini, Corrado, "Tables of Random Permutations by Lincoln E. Moses, Robert V. Oakford," *Journal of the American Statistical Association*, vol.58, no.303, pp.870-871, Sep. 1963. Article (CrossRef Link)

Joonyoung Ahn received M.S. degree in Electrical, Electronic, and Computer Engineering from Sungkyunkwan University, Seoul, Korea, and Ph.D. Candidate in IT Policy and Management from Soongsil University, Seoul, Korea. Mr. Ahn is currently a director in the public devision at TmaxSoft. He is interested in system software, AI, blockchain and web3.

Kijung Ryu received M.S. degree Computer Engineering from Soongsil University Graduate School of Information Science. He has Ph.D. degree in IT Policy and Management from Soongsil University. Mr. Ryu is currently working as a Director of Research Institute at DSTi. He is interested in object-oriented programming, AI, virtual reality

Changik Oh received M.S. degree in Information and Communication Engineering from Sungkyunkwan University, Korea, in 2013 and he is currently attending a Ph.D. Candidate in IT Policy and Management from Soongsil University, Korea. Mr. Oh is working on the information and communication infrastructure and cyber security budget at the Ministry of National Defense of Korea. He is interested in artificial intelligence, intelligent government, information business management, digital innovation, and metaverse.

Taekryong Han received the B.S. degree in Industrial engineering from HanYang University, M.S. degree in Information Protection from Korea University, and Ph.D. degree in Computer Science from Soongsil University, Korea, in 2000, 2020 and 2023, respectively. Mr. Han is currently working as the head of the IT Planning Department at Hyundai Marine & Fire Insurance. He is interested in artificial intelligence, AI call centers, voice bots, robotic process automation, and information security.

Heewon Kim received the B.S. in Electonic Engineering from Seoul University, Korea, M.S. in Electonic Engineering from Seoul University, Korea, and Ph.D. degree in Electonic Engineering from Seoul University, Korea, respectively. Dr. Kim is currently a Professor in the Global School of Media at Soongsil University. He is interested in deep learning, machine learning, computer vision, image processing.

Dongho Kim received the B.S. in Electonic Engineering from Seoul University, Korea, M.S. in Electrical and Electronic Engineering from KAIST, Korea, and Ph.D. degree in Computer Science from George Washington University, respectively. Dr. Kim is currently a Professor in the Global School of Media at Soongsil University. He is interested in virtual reality, the metaverse, and sports convergence, encompassing a broad range of digital content.